

CLAIMS

What is claimed is:

- 5 1. In an electronic device, a method of accessing a library function in a shared library from a dynamic environment, comprising the steps of:
 processing a header file of the library function to extract information for creating an interface to the library function; and
 creating the interface to the library function in the shared library.
- 10
2. The method of claim 1, wherein processing the header file comprises automatically defining parameters for the interface to the library function based on the header file.
3. The method of claim 2, wherein automatically defining parameters for the interface
- 15 to the library function based on the header file comprises creating at least one data structure having selected parameters for interfacing with the library function.
4. The method of claim 1, wherein processing the header file comprises automatically ensuring inputs to the interface to the library function are in the form of selected data
- 20 types based on the header file.
5. The method of claim 4, wherein automatically ensuring inputs to the interface to the library function are in the form of selected data types based on the header file comprises the interface converting data types to the selected data types for the library function.
- 25
6. The method of claim 1, wherein the shared library comprises one of a DLL file and a .so file.
7. The method of claim 1, wherein the header file comprises a C header file.
- 30

8. The method of claim 1, further comprising saving the interface to the library function in the shared library in the dynamic environment in an executable form for subsequent use.
- 5 9. The method of claim 1, further comprising receiving a command to call the library function.
10. The method of claim 1, further comprising executing the library function using the interface from the shared library.
- 10 11. The method of claim 1, wherein the dynamic environment comprises at least one of a text-based modeling application and a graphical-based modeling application.
12. In an electronic device, a method of accessing a library function in a shared library
15 from a dynamic environment, comprising the steps of:
loading the library function from the shared library;
automatically generating an interface to the library function; and
executing the library function using the interface from the shared library.
- 20 13. The method of claim 12, wherein automatically generating the interface comprises the electronic device processing a header file of the library function and extracting information for creating the interface to the library function in the shared library.
- 25 14. The method of claim 13, wherein processing the header file comprises automatically defining parameters for the interface to the library function based on the header file.
15. The method of claim 14, wherein automatically defining parameters for the interface to the library function based on the header file comprises the electronic device creating at least one data structure having selected parameters for interfacing with the library
30 function.

16. The method of claim 12, wherein processing the header file comprises automatically ensuring inputs to the library function are in the form of selected data types based on the header file.
- 5 17. The method of claim 16, wherein automatically ensuring inputs to the library function are in the form of a selected data type based on the header file comprises the interface converting data types to the selected data types for the library function.
- 10 18. The method of claim 12, wherein the shared library comprises one of a DLL file and a .so file.
19. The method of claim 12, wherein the header file comprises a C header file.
- 15 20. The method of claim 12, further comprising saving the interface to the library function in the shared library in the dynamic environment in an executable form for subsequent use.
- 20 21. The method of claim 12, wherein the dynamic environment comprises at least one of a text-based modeling application and a graphical-based modeling application.
22. An electronic device, comprising:
an application providing a dynamic environment;
a shared library accessible by the dynamic environment;
an automated processing function for automatically extracting information for
25 creating an interface to a library function to enable execution of the library function from the shared library.
- 30 23. The device of claim 22, wherein the shared library comprises one of a DLL file and a .so file.
24. The device of claim 22, wherein the header file comprises a C header file.

25. The device of claim 22, further comprising the interface to the library function being saved in the shared library of the dynamic environment in an executable form.

26. The device of claim 22, wherein the dynamic environment comprises at least one of
5 a text-based modeling application and a graphical-based modeling application.

27. A medium for use in a modeling and execution environment on an electronic device,
the medium holding instructions executable using the electronic device for performing a
10 method of accessing a library function in a shared library from a dynamic environment,
the method comprising the steps of:

processing a header file of the library function to extract information for creating
an interface to the library function; and

creating the interface to the library function in the shared library.

15

28. The medium of claim 27, wherein processing the header file comprises
automatically defining parameters for the interface to the library function based on the
header file.

29. The medium of claim 28, wherein automatically defining parameters for the
interface to the library function based on the header file comprises creating at least one
data structure having selected parameters for interfacing with the library function.

30. The medium of claim 27, wherein processing the header file comprises
25 automatically ensuring inputs to the interface to the library function are in the form of
selected data types based on the header file.

31. The medium of claim 30, wherein automatically ensuring inputs to the interface to
the library function are in the form of selected data types based on the header file
30 comprises the interface converting data types to the selected data types for the library
function.

32. The medium of claim 27, wherein the shared library comprises one of a DLL file and a .so file.

33. The medium of claim 27, wherein the header file comprises a C header file.

5

34. The medium of claim 27, further comprising saving the interface to the library function in the shared library in the dynamic environment in an executable form for subsequent use.

10 35. The medium of claim 27, wherein the dynamic environment comprises at least one of a text-based modeling application and a graphical-based modeling application.